An Intuitive Framework for Real-Time Freeform Modeling

Mario Botsch Leif Kobbelt

Computer Graphics Group
RWTH Aachen University

Abstract

We present a freeform modeling framework for unstructured triangle meshes which is based on constraint shape optimization. The goal is to simplify the user interaction even for quite complex freeform or multiresolution modifications. The user first sets various boundary constraints to define a custom tailored (abstract) basis function which is adjusted to a given design task. The actual modification is then controlled by moving one single 9-dof manipulator object. The technique can handle arbitrary support regions and piecewise boundary conditions with smoothness ranging continuously from C^0 to C^2. To more naturally adapt the modification to the shape of the support region, the deformed surface can be tuned to bend with anisotropic stiffness. We are able to achieve real-time response in an interactive design session even for complex meshes by precomputing a set of scalar-valued basis functions that correspond to the degrees of freedom of the manipulator by which the user controls the modification.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Physically based modeling; I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction Techniques

Keywords: surface editing, freeform design, user interaction

1 Introduction

Computer aided geometric design techniques have become an important key technology in the industrial design and development process. The availability of digital 3D models during the various stages of the design process triggers a large number of applications ranging from numerical simulation and assembly planning to design studies and product presentation. This variety of applications implies that there are many different (and sometimes contradicting) requirements for how these 3D models have to be accessed in an interactive design session. Especially in the earlier (conceptual) design stages, the major bottleneck is still the freeform design metaphor which should allow the user to convey an imaginary shape to the computer system in an intuitive fashion.

The fundamental problem here is that the space of possible geometric shapes is extremely high dimensional and has a very complex structure with esthetically pleasing shapes sometimes lying surprisingly close to unacceptable shapes. The designer has to explore this rich space of shapes by just pressing buttons or clicking and dragging 2D positions on the screen. More sophisticated interaction technology like haptic input devices, immersive displays [Schkolne et al. 2001], or two-handed input metaphors [Llamas et al. 2003] are available today but they did not replace the well-established PC-based working place as it can be found in every industrial design company.

In an interactive design session, the user usually drags control vertices which have three degrees of freedom (translation) or he moves more general manipulator objects having six (translation and rotation) or nine (plus scaling) degrees of freedom. For complex shape modifications several control handles have to be dragged sequentially or simultaneously. This rudimentary interface is the reason why it takes highly skilled experts to operate a CAD system. Nevertheless it is the widely accepted standard and implemented in many NURBS or subdivision based modeling frameworks.

Let S be the given shape that the designer wants to modify into another shape S' and let us assume that S and S' do not differ too much. Obviously, every extreme modification can be decomposed into a sequence of smaller modifications and this is what designers usually do – even traditional designers working with real clay, not just CAD designers.

If we accept that interactive design is about dragging manipulators and verifying the result by visual feedback then any shape modification can be characterized by

$$S' = S + B(\delta C) \tag{1}$$

where B represents an abstract basis function and δC somehow represents the change of position and orientation of the control handles. For a modeling tool based on NURBS surfaces, e.g., B could be the set of tensor-product basis functions and δC the displacement vectors by which we shift the corresponding control vertices.

A shape modification is complex if the update $B(\delta C)$ is complex. This can be achieved by either providing a complicated handle object with many degrees of freedom or by using special basis functions that are adapted to the desired modification. Since our goal is to keep the user interaction simple, the possible types of modifications that we can apply to the object S are hence limited by the abstract basis functions that our system associates with the control handles.

In the case of NURBS or subdivision surfaces, this means that with each elementary modification we can add a smooth bump with rectangular or polygonal support to the surface. Every more sophisticated modeling operation has to be built from these elementary modifications.

An important limitation with most of the existing modeling frameworks is that the underlying mathematical surface representation is tightly linked to the type and number of control handles. This problem is more than obvious if our geometry representation is based on unstructured triangle meshes since here, shifting a (control) vertex just adds a tiny hat function to the surface. In order to perform some non-trivial modification one has to manually move a larger number of control vertices simultaneously. In the setting of (1) this would correspond to a highly complex δC, i.e. a highly complex user interaction.

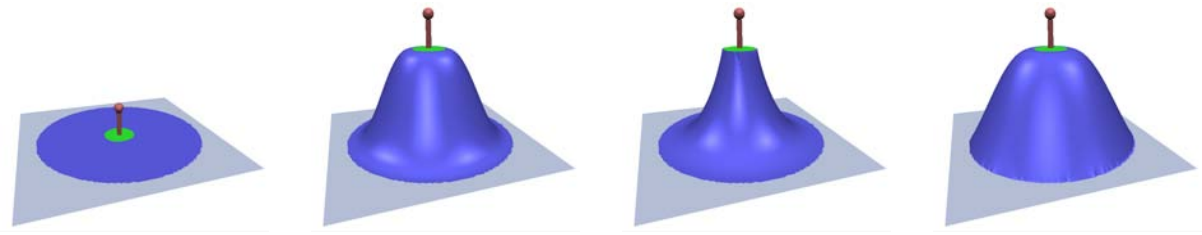

Figure 1: In our modeling metaphor, we define a custom tailored basis function by selecting a support region (blue) and a handle region (green). The smoothness conditions at the inner and outer boundary can be controlled independently and continuously blended between C^0 and C^2. From left to right we show the initial configuration, C^2 at inner and outer boundary, C^0 at inner and C^2 at outer, as well as C^2 at inner and C^0 at outer boundary. The blue, green and gray regions correspond to the sets of vertices **p**, **h**, and **f** in equation (5), respectively.

This is why for polygon meshes, control handle based modification metaphors have been developed which are mostly independent from the underlying tessellation of the surface. This means we simplify the structure of C in (1) by making B slightly more complicated.

Freeform deformation [Sederberg and Parry 1986; Coquillart 1990; MacCracken and Joy 1996] is probably one of the most prominent examples where shifting a control vertex in a spatial grid causes a deformation of the embedding space around a 3D model and thereby induces a global modification on the model itself. While this is a very intuitive modeling metaphor, is does not provide significantly more degrees of freedom (bumps over simple support regions). Moreover, the support of the modification is sometimes difficult to predict as it is determined by intersecting a volumetric basis function's support with the modified surface.

The purpose of this short paper is to describe another modeling metaphor which provides the maximum flexibility with respect to the set of potential basis functions. The idea is to help the designer to define his own custom tailored basis function that is optimally adapted to the intended modification. Then this basis function is associated with a manipulator object that the user can move interactively to do the actual shape editing operation in real-time.

Our goal is to provide as many degrees of freedom as possible for the definition of the basis function B to offer enough flexibility for non-trivial modifications. Yet, we always keep in mind that the resulting shape modeling system should be simple and intuitive enough for an average (not specialized) user. Hence, after the definition of the boundary constraints, the user interaction is restricted to moving a single manipulator object C.

While we are integrating various known geometry processing concepts into our freeform modeling framework, we also extend these techniques to meet the central requirements of our modeling system. The two major innovations are that we obtain flexible shape control by using an anisotropic discretization of the energy functional which determines how the surface bends under deformation and that by precomputing a set of linear basis functions, we easily achieve real-time feedback even when modifying large surface areas.

2 The modeling metaphor

In Eq. (1) we represented an arbitrary freeform modification by an abstract basis function B which is added to an existing shape S. In order to specify this basis function for a particular modification we have to define its *support*, i.e., the region of the surface S that should be affected by this modification and its *characteristic shape*.

Flexibility with respect to the support of B means that we can select an arbitrary region, convex or non-convex with smooth or non-smooth boundary, and which can be aligned to any feature on the surface. A simple way to let the user define this region is by letting him draw directly on the surface either the outline of the region or the complete region (with some painting tool).

The characteristic shape of the basis function B is most intuitively defined by terms like *smoothness*, *stiffness*, or *fullness*. Here, smoothness is a property that describes how the deformed part of the surface connects with the unmodified part, stiffness or tension describes how the curvature is distributed (equally distributed vs. clustered near the constraints), and fullness rates the relation between height and volume of the basis function (pointed vs. blobby).

To control stiffness and smoothness, we let the user select either the interior of the support region or (a segment) of its boundary and then provide a simple slider to increase or decrease the respective parameter. Even if the mathematical meaning of these terms might not be obvious for the non-expert user, the visual feedback when moving the slider still allows him to quickly set the corresponding scalar parameters according to his design intend.

In order to map the control of the modification to a 9-dof manipulator object, the user selects another region, the *handle region*, in the interior of the support region (cf. Fig. 1, left). The manipulator is then rigidly attached to this surface patch and hence moving the manipulator moves the surface patch accordingly. The remaining part of the surface, i.e., support region minus handle region is supposed to smoothly bend according to the translation, rotation and scaling of the handle region. The *fullness* of the basis function B can hence be controlled by the size and the shape of the handle region.

The modeling metaphor as we described it so far, i.e. the definition of an abstract basis function B which is then controlled by a simple manipulator object, provides a flexible tool for freeform shape editing. We can easily integrate this metaphor into a multiresolution modeling framework in order to preserve the local detail information of an object when applying a global modification.

For this we need a decomposition operator which separates the high-frequency detail from the low-frequency global shape. The freeform modeling technique is then applied to the low-frequency component and finally the detail information is added back to the modified surface by a reconstruction operator. The multiresolution decomposition and reconstruction can be hidden from the user such that he seems to interact with the detailed surface while the frequency of the modification is controlled by the size of the support region.

Figure 2: The order k of the energy functional defines the stiffness of the surface in the support region and the maximum smoothness C^{k-1} of the boundary conditions. From left to right: membrane surface ($k=1$), thin-plate surface ($k=2$), minimal curvature variation ($k=3$).

3 The mathematical realization

The mathematical techniques that we need in order to implement the design metaphor described in the last section have to be flexible enough to allow for arbitrary support and characteristics but they also have to be efficient enough to give real-time responses when the user moves the manipulator.

Smooth deformation of a surface with respect to boundary conditions is most elegantly modelled by an energy minimization principle [Moreton and Sequin 1992; Welch and Witkin 1992; Kobbelt et al. 1998; Du and Qin 2000]. The surface is assumed to behave like a physical skin which stretches and bends as forces are acting on it. Mathematically this behavior can be captured by an energy functional which penalizes stretch or bending. Then the optimal surface is the one that minimizes this energy while satisfying all the prescribed boundary conditions. The advantage of this formulation is that it allows us to take arbitrary boundary conditions into account and the optimal solution is known to have certain smoothness properties. When changing the boundary conditions, the optimal surface changes accordingly and this is why we call this approach *boundary constraint modeling* (BCM).

An alternative approach which mimics boundary constraint modeling is to compute a blending function that propagates the constraints into the interior of the support region [Singh and Fiume 1998; Bendels and Klein 2003; Pauly et al. 2003]. While these methods are extremely efficient, they usually do not provide the full generality since they often assume a certain topology or shape of the support region.

Linear system derivation

For efficiency reasons the energy functionals that are used most often are quadratic functionals with the generic form [Kobbelt 1997]

$$E_k(S) = \int F_k\left(S_{u\ldots u}, S_{u\ldots uv}, \ldots, S_{v\ldots v}\right) \tag{2}$$

where the expressions S_* stand for the partial derivatives of order k with respect to a surface parametrization $S:\Omega\rightarrow\mathbb{R}^3$ which is locally as close as possible to isometric.

In order to actually compute the solution to the above optimization problem one usually applies variational calculus to derive the corresponding Euler-Lagrange equation which characterizes the minimizers of (2). For the most common quadratic energy functionals, the resulting linear differential equation has the form

$$\begin{aligned} \Delta^k S(x) &= 0, && x\in\Omega\setminus\delta\Omega \\ \Delta^j S(x) &= b_j(x), && x\in\delta\Omega,\ j<k \end{aligned} \tag{3}$$

where Δ is the Laplace operator and the boundary constraints b_j of order $j<k$ on $\delta\Omega$ imply a non-trivial solution. For $k=1$ this equation characterizes *membrane surfaces* which minimize surface area, for $k=2$ it characterizes *thin plate surfaces* which minimize surface bending and for $k=3$ we obtain surfaces that minimize the variation of linearized curvature (cf. Fig. 2). Higher order equations are usually not recommended because of numerical instabilities.

The types of boundary conditions that we can impose on (3) can be up to C^{k-1}. Hence for $k\geq 2$ we can choose between "hinged" boundaries ($C^0,\ j=0$) and "clamped" boundaries ($C^1,\ j=1$). Moreover we can continuously blend boundary conditions between C^0 and C^{k-1} with a smoothness parameter $c(p)\in[0,k-1]$ for constrained boundary vertices p by modifying the recursive definition of the higher order Laplacian from [Kobbelt et al. 1998] to be

$$\begin{aligned} \bar{\Delta}^k(p) &:= \Delta\left(\lambda_{k-1}(p)\cdot\bar{\Delta}^{k-1}(p)\right) \\ \lambda_k(p) &:= \begin{cases} 1, & c(p)>k \\ c(p)-k, & k-1\leq c\leq k \\ 0, & c(p)<k-1 \end{cases} . \end{aligned}$$

Since we want to use a triangle mesh as the underlying surface representation, we have to discretize the Laplace operator [Desbrun et al. 1999; Meyer et al. 2003] by

$$\Delta(p_i) := \frac{2}{A(p_i)} \sum_{p_j\in N(p_i)} \left(cot\alpha_{ij} + cot\beta_{ij}\right)\left(p_j - p_i\right), \tag{4}$$

where $\alpha_{ij}=\angle(p_i,p_{j-1},p_j)$ and $\beta_{ij}=\angle(p_i,p_{j+1},p_j)$ for a vertex p_i and its one-ring neighbors p_j and $A(p_i)$ denotes the Voronoi area around the vertex p_i. By this (3) becomes a sparse linear system

$$\left(\begin{array}{c|c} \bar{\Delta}^k & \\ \hline 0 & I_{F+H} \end{array}\right) \left(\begin{array}{c} \mathbf{p} \\ \mathbf{f} \\ \mathbf{h} \end{array}\right) = \left(\begin{array}{c} \mathbf{0} \\ \mathbf{f} \\ \mathbf{h} \end{array}\right), \tag{5}$$

where $\mathbf{p}=(p_1,\ldots,p_P)$ is the vector of free vertices in the interior of the support region, $\mathbf{f}=(f_1,\ldots,f_F)$ are the fixed vertices outside the support region and $\mathbf{h}=(h_1,\ldots,h_H)$ are the vertices inside the handle region. These sets of vertices correspond to the *blue*, *gray* and *green* surface regions shown in Fig. 1, respectively. Since $\mathbf{f}$ and $\mathbf{h}$ are fixed, they impose the boundary conditions on the system. Notice that only $k+1$ rings of fixed vertices are used to prescribe C^k boundary constraints. For the sake of simplicity we combined the optimality conditions and the boundary conditions into one equation. In the following we refer to the matrix in (5) as L.

When the user moves the handle region, the vertices in $\mathbf{h}$ change their position and provide a new right hand side for the linear system. By solving (5) again we hence compute the vertex positions in $\mathbf{p}$ as a linear function of $\mathbf{h}$.

Anisotropic bending

If we use the standard discretization (4) of the Laplace operator then the resulting optimal surface bends isotropically even if the support region is anisotropic. Since the shape of the handle and support regions are considered as design parameters when defining the basis function for a particular modification, we would rather like the resulting surface to better adapt its bending behavior to the boundary conditions. As shown in Fig. 3 the basis function looks more natural if the impact of the handle region is propagated through the support region in such a way that its "iso-contours" hit the outer boundary everywhere with approximately the same slope.

This can be achieved by discretizing the Laplace operator (4) not with respect to the mesh itself (Laplace-Beltrami) but rather with respect to a special parametrization that "factors out" the anisotropy. First we compute a conformal parametrization for the support region [Lévy et al. 2002; Ray and Levy 2003]. This yields a planar triangulation with the same connectivity as the support region. Then we apply a principal axis transform and scale this planar triangulation along its principal axes such that its diameter is approximately the same in each direction. On this scaled triangulation we finally compute the weights for the Laplace operator that we then use in (5). Fig. 3 shows the effect of minimizing this anisotropic energy functional.

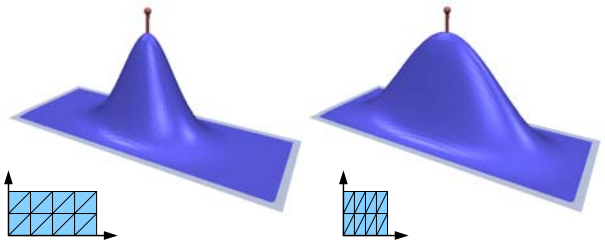

Figure 3: Isotropic (left) and anisotropic basis functions (right) with the corresponding parameter domains over which the Laplace operator is discretized.

Precomputed basis functions

The technique as we described it so far requires to solve a linear system for the free vertex positions whenever the manipulator (and hence the handle region) moves. Although we can use a highly efficient multi-grid solver for this task we still do not achieve a sufficiently high frame rate, especially when minimum curvature variation surfaces ($k = 3$) are computed and the number of vertices in the support region is on the order of 10^4 or higher.

By precomputing a special set of basis functions that directly correspond to the degrees of freedom of the manipulator, we can significantly reduce the per-frame computing costs. Since the solution of (5) can be expressed explicitly in terms of the inverse matrix L^{-1}, this set of basis functions is represented by (a combination of) column vectors of L^{-1}. A similar technique was used in the physically-based modeling system ArtDefo [James and Pai 1999], where also a set of column vectors of an inverse matrix is precomputed in order to speed up the surface updating.

First we observe that the explicit solution of (5) is

$$\begin{pmatrix} \mathbf{p} \\ \mathbf{f} \\ \mathbf{h} \end{pmatrix} = L^{-1} \begin{pmatrix} \mathbf{0} \\ \mathbf{f} \\ \mathbf{h} \end{pmatrix} = L^{-1} \begin{pmatrix} \mathbf{0} \\ \mathbf{f} \\ \mathbf{0} \end{pmatrix} + L^{-1} \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{h} \end{pmatrix} \quad (6)$$

with the first term on the right hand side being constant and the second term depending on the vertices in the handle region.

During interactive shape editing we are using a 9-dof manipulator which provides an intuitive interface to control an affine map T that is applied to the handle region. If we pick four affinely independent vertices a, b, c, and d from the handle region then these define an affine frame and there exists a matrix $\mathbf{Q} \in \mathbb{R}^{H\times 4}$ of affine combinations such that

$$\mathbf{h} = \mathbf{Q}\,[a,b,c,d]^T.$$

Due to affine invariance, applying an affine map T (controlled by the manipulator) to the handle vertices $\mathbf{h}$, is then equivalent to applying T to the affine frame, i.e.

$$T(\mathbf{Q}[a,b,c,d]^T) = \mathbf{Q}\,T([a,b,c,d]^T).$$

As a consequence we can rewrite the non-constant term in (6) as

$$\begin{pmatrix} \mathbf{p} \\ \mathbf{f} \\ \mathbf{h} \end{pmatrix} = L^{-1} \begin{pmatrix} \mathbf{0} \\ \mathbf{f} \\ \mathbf{0} \end{pmatrix} + L^{-1} \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{Q} \end{pmatrix} [a,b,c,d]^T.$$

This however means that we only have to solve the system (5) in a pre-processing step for 7 different right hand sides: the three columns of $[\mathbf{0},\mathbf{f},\mathbf{0}]^T$ and the four columns of $[\mathbf{0},\mathbf{0},\mathbf{Q}]^T$. The latter allows us to precompute the "basis function" matrix $B = L^{-1}[\mathbf{0},\mathbf{0},\mathbf{Q}]^T$. Then for every frame we simply apply the manipulator transformation to the four handle points a, b, c, and d and add the current displacement vectors $B\,[a,b,c,d]^T$ to the constant part in (6).

4 Results

We integrated our freeform modeling metaphor into a multiresolution mesh editing framework like the one described in [Kobbelt et al. 1998]. The decomposition and editing operators are both based on BCM and the representation of high-frequency detail is implemented in terms of normal displacement vectors [Kobbelt et al. 1999].

A real world example is shown in Fig. 5, where the sillboard of a car is to be lowered. Exploiting the flexibility provided by continuous boundary smoothness avoids the generation of an unwanted point of inflection along the feature line and the anisotropic Laplace discretization propagates the displacement naturally over the support region. Features are preserved due to multiresolution decomposition and reconstruction (hidden from the user).

Notice that multiple independent handle regions each controlled by their own manipulator object are no problem for the setup described

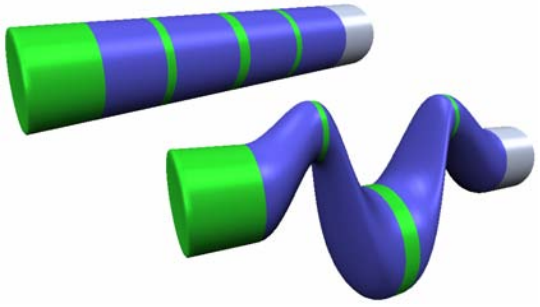

Figure 4: Example of a non-disc shaped modification using multiple handles. The left cap (green) is defined as a handle component that is not to be moved. The actual modification is done by defining additional handle components as rings (green).

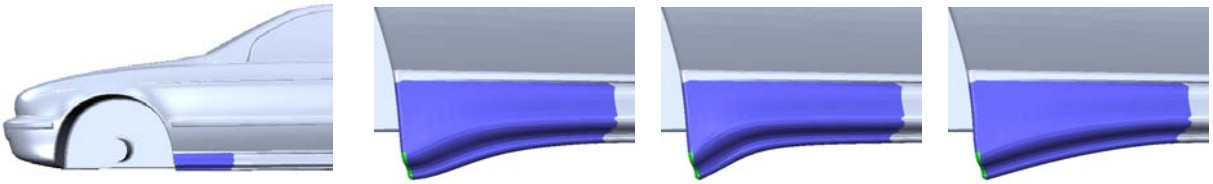

Figure 5: Multiresolution modification of the sillboard using flexible boundary conditions and anisotropic Laplacian: An unwanted point of inflection (center left) is avoided by reducing the smoothness constraint near the handle region to C^0 continuity. Switching from the isotropic discretization of the Laplace operator (center right) to the anisotropic one finally leads to the intended natural displacement propagation.

in this paper. All we have to do is to split **h** into several components and precompute the corresponding four basis functions for each of them. By this it is even possible to generate modifications with a support region that is not topologically equivalent to a disk. We simply exclude some regions from the deformable area by labelling them as special handle regions that cannot be moved (cf. Fig. 4). More complex support regions can be used if we restrict to the isotropic Laplacian because this avoids the parametrization step (cf. Sect. 3).

The last example shows a complex modification of a car's hood (cf. Fig. 6, 250k triangles). Here we use multiple handle regions placed at the wheel houses and the grill, enabling us to stretch the hood while keeping the wheel houses circular. The support region of this modification contains 35k vertices, the complete precomputation (multigrid hierarchy, basis functions, multiresolution hierarchy) took less than 15s. The actual surface editing can be done with 12 fps, where only 25ms are required to compute the BCM surface and the remaining time is used for detail reconstruction and rendering.

In an industrial evaluation the presented modeling system proved to be both sufficiently flexible as well as very intuitive, enabling also the non-experts to perform their desired shape modifications. Due to the precomputed basis functions, deformations even on complex models could be performed in real-time.

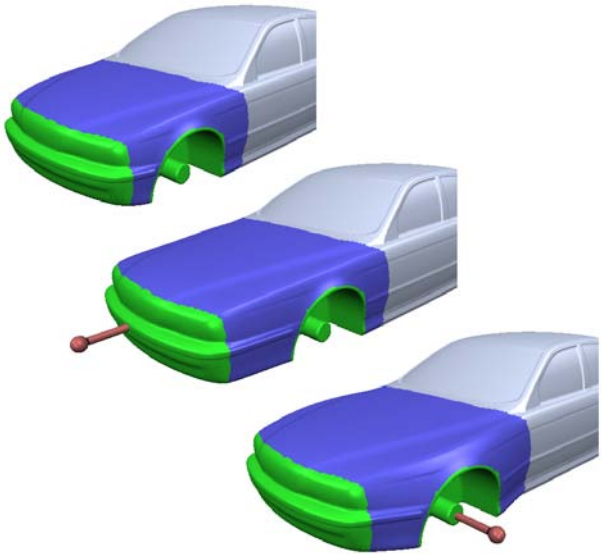

Figure 6: Stretching the hood using multiple independent handle components. Notice that rigidly preserving the circular shape of the wheel houses would be very difficult using a volumetric deformation tool like freeform deformation.

References

BENDELS, G. H., AND KLEIN, R. 2003. Mesh forging: editing of 3D-meshes using implicitly defined occluders. In *Proceedings of the Eurographics/ACM SIGGRAPH symposium on Geometry processing*, Eurographics Association, 207–217.

COQUILLART, S. 1990. Extended free-form deformation: a sculpturing tool for 3D geometric modeling. In *Computer Graphics (Proceedings of ACM SIGGRAPH 90)*, ACM, 187–196.

DESBRUN, M., MEYER, M., SCHRÖDER, P., AND BARR, A. H. 1999. Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proceedings of ACM SIGGRAPH 99*, ACM Press/ACM SIGGRAPH, 317–324.

DU, H., AND QIN, H. 2000. Direct manipulation and interactive sculpting of pde surfaces. In *Proceedings of Eurographics 00*, 261–270.

JAMES, D. L., AND PAI, D. K. 1999. ArtDefo: accurate real time deformable objects. In *Proceedings of ACM SIGGRAPH 99*, ACM Press/ACM SIGGRAPH, 65–72.

KOBBELT, L., CAMPAGNA, S., VORSATZ, J., AND SEIDEL, H.-P. 1998. Interactive multi-resolution modeling on arbitrary meshes. In *Proceedings of ACM SIGGRAPH 98*, ACM Press/ACM SIGGRAPH, 105–114.

KOBBELT, L., VORSATZ, J., AND SEIDEL, H.-P. 1999. Multiresolution hierarchies on unstructured triangle meshes. *Computational Geometry: Theory and Applications 14*.

KOBBELT, L. 1997. Discrete Fairing. In *Proceedings on 7th IMA Conference on the Mathematics of Surfaces*, 101–131.

LÉVY, B., PETITJEAN, S., RAY, N., AND MAILLOT, J. 2002. Least squares conformal maps for automatic texture atlas generation. *ACM Transactions on Graphics 21*, 3, 362–371.

LLAMAS, I., KIM, B., GARGUS, J., ROSSIGNAC, J., AND SHAW, C. D. 2003. Twister: a space-warp operator for the two-handed editing of 3D shapes. *ACM Transactions on Graphics 22*, 3, 663–668.

MACCRACKEN, R., AND JOY, K. I. 1996. Free-form deformations with lattices of arbitrary topology. In *Proceedings of ACM SIGGRAPH 95*, ACM Press/ACM SIGGRAPH, 181–188.

MEYER, M., DESBRUN, M., SCHRÖDER, P., AND BARR, A. H. 2003. Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and Mathematics III*, H.-C. Hege and K. Polthier, Eds. Springer-Verlag, Heidelberg, 35–57.

MORETON, H., AND SEQUIN, C. 1992. Functional optimization for fair surface design. In *Computer Graphics (Proceedings of ACM SIGGRAPH 92)*, 167–176.

PAULY, M., KEISER, R., KOBBELT, L. P., AND GROSS, M. 2003. Shape modeling with point-sampled geometry. *ACM Transactions on Graphics 22*, 3, 641–650.

RAY, N., AND LEVY, B. 2003. Hierarchical Least Squares Conformal Map. In *Proceedings of Pacific Graphics 03*, 263–270.

SCHKOLNE, S., PRUETT, M., AND SCHRÖDER, P. 2001. Surface drawing: creating organic 3D shapes with the hand and tangible tools. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM Press, 261–268.

SEDERBERG, T. W., AND PARRY, S. R. 1986. Free-form deformation of solid geometric models. In *Computer Graphics (Proceedings of ACM SIGGRAPH 86)*, 151–159.

SINGH, K., AND FIUME, E. 1998. Wires: A geometric deformation technique. In *Proceedings of ACM SIGGRAPH 98*, ACM Press/ACM SIGGRAPH, 405–414.

WELCH, W., AND WITKIN, A. 1992. Variational surface modeling. In *Computer Graphics (Proceedings of ACM SIGGRAPH 92)*, 157–166.